

Partial English Translation of Japanese Laid-Open Patent
Application NO.4-359315

[Title of the Invention]

Data compression controlling apparatus and data
restoration controlling apparatus

Abstract

[Abstract]

[Purpose]

A data compression controlling apparatus is provided wherein the optimum data compression method is automatically selected depending on the input data thus increasing the compression rate. Also, a data restoration controlling apparatus is provided wherein the optimum data restoring method is automatically selected for restoration of data.

[Constitution]

An input data received at a buffer memory 3 is compressed by the first and the second compression method in two data compressors 4 and 6 respectively and then saved in their corresponding buffer memories 5 and 7. The data saved in the buffer memories 3, 5, and 7 are measured in the data length by three counters 8, 9, and 10 respectively and compared by a compression rate comparator 11. From a comparison result, the compression method providing the highest rate of the data compression is selected and notified to a data compression method switch 12 which in turn allows the data compression with the selected compression method. An attribute data indicating

the data compression method used is generated by a file attribute data generator 13 and recorded/played back by a record/playback module 16. In a data restoration mode, the data restoration method is selected by a data restoration method switch 23 referring to the attribute data.

[0015]

[Description of Examples]

A data compression controlling apparatus and a data restoration controlling apparatus according to the present invention will be described for use in a magnetic record/playback apparatus, referring to the relevant drawings. Fig. 1 is a block diagram of a data record/playback apparatus showing the data compression controlling apparatus of a first example of the present invention. Fig. 2 is a block diagram of the data record/playback apparatus showing the data compression controlling apparatus of the first example of the present invention.

[0016]

The description starts with defining some terms used herein. "File" is a group of related data provided as a unit to be transferred from a host computer for backup of the data, "Block" is a unit of data divided from the file. "Block header" is a preamble data attached to each block and having information about the block. "Record" is a minimum unit of data for data transfer to and from a record/playback module and data storage on e.g. a recording tape. The records are classified into fixed length records of which the length is fixed and variable length

records of which the length is variable.

[0017]

As shown in Fig. 1, an input/output module 1 is provided for receiving a data from a host computer (not shown) or transferring to the host computer a data restored from a tape medium while serving as a data inputting means. A file data separator 2 separates a file data received from the input/output module 1 into block of a predetermined length. A buffer memory 3 temporarily saves a data received from the file data separator 2. A data compressor 4 compresses a data received from the buffer memory 3 with the use of a first compression algorithm. A buffer memory 5 temporarily saves a data received from the data compressor 4. A data compressor 6 compresses a data receive from the buffer memory 3 wit the use of a second compression algorithm. A buffer memory 7 temporarily saves a data received from the data compressor 6. A counter 8 is a first counting means for measuring the length of a pre-compression data saved in the buffer memory 3. A couple of counters 9 and 10 are provided as a second counting means for measuring the length of post-compression data saved in the buffer memories 5 and 6 respectively. A compression rate comparator 11 compares measurements of the three counters 8, 9, and 10 to examine which compression method is higher in the compression rate. A data compression method switch 12 controls turning on and off of the data compression function of the data compressors 4 and 6 in response to a command output of the compression rate comparator 11. A file attribute data generator 13 generates a file

attribute data on a buffer memory 14 for recording a file management data about files recorded on the tape medium. The buffer memory 14 temporarily saves management data including the file attribute data and a block attribute data to be recorded on the tape medium and also holds a management data received from the tape medium. A buffer memory selector 15 selects one of the buffer memories 3, 5, and 7 in response to a buffer select signal supplied from the compression rate comparator 11 for transferring the data to a record/playback module 16. Also, the file attribute data generated on the buffer memory 14 during the file management data recording process is received by the record/playback module 16.

[0018]

The record/playback module 16 records on a tape medium a data received via the buffer memory selector 15 or plays back a data from a tape medium. The record/playback module 16 consists mainly of a drive mechanism and a record/playback controller as not explained herein in more detail. The drive mechanism drives a tape medium and records or plays back a data on the tracks of the tape medium with its head. The record/playback controller carries out an action of signal processing a data received. In a record mode, the controller performs, for example, an action of attaching an error correction code and actuates the drive mechanism while providing the head with a recording current for producing the tracks on the tape medium. In a playback mode, the controller reads out a desired data with the head and subjects the data

of playback signal to an inverse of the signal processing action in the record mode. A controller 17 controls the input and output of data on the host computer and the entire actions of the data compression controlling apparatus and the data record/playback apparatus.

[0019]

In the data record/playback apparatus with the data restoration controlling apparatus shown in Fig. 2, the input/output module 1 receives a data from a host computer (not shown) and transfers a data restored from a tape medium to the host computer. The record/playback module 16 records a data onto a tape medium and restores the data from the tape. A buffer memory 18 temporarily saves a data received from the record/playback module 16. A data restorer 19 restores a data compressed with the first compression algorithm and receive from the buffer memory 18. A data restorer 20 restores a data compressed with the second compression algorithm and receive from the buffer memory 18. A buffer memory 21 temporarily saves a data restored by the data restorer 19 or 20. A file attribute data detector 22 examines a directory data read out from a tape medium and transferred by the buffer memory 18 to determine the attribute of a file. A data restoration method switch 23 upon receiving a command output of the file attribute data detector 22 controls turning on and off of the data restoring action of the two data restorers 19 and 20. In response to a buffer selection signal from the data restoration method switch 23, a buffer memory selector 24 selects either the buffer memory

18 or a buffer memory 21 for transferring a data to the input/output module 1. The controller 17 controls the input and output of control data on the host computer and the entire actions of the data restoration controlling apparatus and the data record/playback apparatus.

[0020]

It is assumed that dictionary buffer memories are commonly needed for the compression and restoration of data and hence installed in the data compressor 4, the data compressor 6, the data restorer 19, and the data restorer 20 but not shown. While the buffer memories 3, 5, 7, and 14 for data compression and the buffer memories 18 and 24 for data restoration are illustrated as separate blocks, they may be assembled to a single unit to allow the common use of any buffer memory portion. Also, while the data compression controlling apparatus and the data restoration controlling apparatus are separately illustrated in Figs. 1 and 2 respectively, they may be combined to one single construction of which any portion can be utilized for common use.

[0021]

The action of the data compression controlling apparatus and the data restoration controlling apparatus having the above arrangements will now be described. The description starts with a procedure of data recording when a file data is not separated into blocks, referring to a flowchart shown in Fig. 3. The file data received from a host computer is transferred from the input/output module 1 to the file data separator 2 (Step

101). As the file data is not subjected to separation into blocks, it is simply passed through the file data separator 2 without being process. The file data remaining unchanged is transferred from the file data separator 2 to the buffer memory 3. The file data is saved in the buffer memory 3 and examined by the counter 8 to count how many bytes it carries in the data length (Step 102).

[0022]

It is herein assumed that the capacity of the buffer memory 3 is 32 kilobytes. All the file data may be saved in the buffer memory 3. Alternatively, if the file data is greater in the volume than the capacity of the buffer memory 3, the buffer memory 3 upon receiving the file data transfers its saved data to the data compressor 4 and the data compressor 6 for data compression. The data compressed by the data compressor 4 is then saved in the buffer memory 5 and simultaneously examined by the counter 9 to determine its data length. Similarly, the data compressed by the data compressor 6 is saved in the buffer memory 7 and simultaneously examined by the counter 10 to determine its data length (Step 103). When the data length of the compressed data saved in the buffer memory 5 or 7 reaches 32 kilobytes, no more data is received by the buffer memory 5 or 7 since both the buffer memories 5 and 7 have a capacity of 32 kilobytes.

[0023]

The compression algorithm may be selected from a Lempel-Ziv algorithm and other appropriate techniques. Characteristic

examples of the compression algorithm are a technique of converting a chain of similar data into its coded form concerning each data string and its length, a technique of assigning a short bit length to a data or a group of data at a higher appearance frequency before conversion to a coded form, and a technique of converting data into its coded form concerning a data length and a pointer for a previous group of data which has repeatedly been obtained. Every technique permits the input data (a stream of bits) to be converted into its coded form with the use of a compression algorithm. Such compression algorithms are well known and will be explained in no more detail. It is free to select any of the known compression algorithms.

[0024]

As all the file data saved in the buffer memory 3 has been compressed by the data compressor 4 and the data compressor 6, measurements of the length of the non-compressed data and the compressed data of the counters 8, 9, and 10 are compared by the compression rate comparator 11 (Step 104).

[0025]

When the compressed data measured by the counter 9 is the smallest in the data length, it is determined that the first compression algorithm is more effective for the data compression and the procedure goes to Step 105 where the compression rate comparator 11 directs the data compression method switch 12 to turn the data compressor 4 on and the data compressor 6 off and also directs the buffer memory selector

15 to select the buffer memory 5. When the compressed data measured by the counter 10 is the smallest in the data length, it is determined that the second compression algorithm is more effective for the data compression and the procedure goes to Step 109 shown in Fig. 4 where the compression rate comparator 11 directs the data compression method switch 12 to turn the data compressor 4 off and the data compressor 6 on and also directs the buffer memory selector 15 to select the buffer memory 7. When the non-compressed data measured by the counter 8 is the smallest in the data length, it is determined that both the first and second compression algorithms are not effective for the data compression. Accordingly, the compression rate comparator 11 directs the data compression method switch 12 to turn the two data compressors 4 and 6 off and also directs the buffer memory selector 15 to select the buffer memory 3 (Step 113).

[0026]

In response to commands from the compression rate comparator 11, the data compression method switch 12 controls the turning on and off of the data compression action of the two data compressors 4 and 6. Simultaneously, the compression rate comparator 11 provides the file attribute data generator 13 with a signal instructing which of the compression algorithms is used and whether the data compression is carried out. In turn, the file attribute data generator 13 assigns a code data of the compression algorithm used to the attribute data in a directory data of the file (Steps 106, 110, and 114). The

directory data will be explained later in more detail.

[0027]

Then, when the first compression algorithm is selected, the data is transferred from the buffer memory 5 via the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 107). In case that the remaining of the file data is received from the host computer, it is saved in the buffer memory 3, compressed by the data compressor 4, passed to the buffer memory 5, and transferred by the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 108).

[0028]

When the second compression algorithm is selected, the data is transferred from the buffer memory 7 via the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 111). In case that the remaining of the file data is received from the host computer, it is saved in the buffer memory 3, compressed by the data compressor 6, passed to the buffer memory 7, and transferred by the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 112).

[0029]

When it is determined that the data compression is hardly effective, the pre-compression data saved in the buffer memory 3 is directly transferred by the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 115). In case that the remaining of the file data is received

from the host computer, it is saved in the buffer memory 3 and directly transferred by the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 116).

[0030]

A procedure of updating the directory on a tape medium will be described referring to a flowchart shown in Fig. 5. When receiving a file mark recording command from the host computer after the transfer of the file data is completed, the controller 17 directs the record/playback module 16 to record a file mark on the tape medium. Then, the file mark is recorded on the tape medium by the record/playback module 16 (Steps 201, 202, and 203).

[0031]

It is adapted in this example to provide the directory at the front end of the tape medium for storage of the attribute data of each file data. The directory can be updated when another file data is added or overwritten. Fig. 6 illustrates a pattern of the data recorded on a tape medium by the action of the data record/playback apparatus of this example. As shown in Fig. 6, the tape medium has from its front end a tape header 31 carrying the ID data (including a tape format data), a directory 32 carrying the management data and the attribute data of each file, data records 33 for storage of the file data, and an EOI (End Of Information) 34 indicating the end of the records which are separated from each other by a gap 35. The data records 33 are isolated as identified with corresponding file

marks 36.

[0032]

Fig. 7(a) illustrates an example of the directory 32. The directory 32 comprises a number of tables, each table composed of 8 bytes and provided for each file. The table for each data file includes a file number data 41 (3 bytes), a file size data 42 (4 bytes), and a file attribute data 43 (1 bytes). The file size data 42 represents the length of the file in bytes. The file attribute data 43 is a code data of the data compression algorithm employed for compressing the file data. More specifically, when the file attribute data 43 is "1", the first compression algorithm is used and when "2", the second compression algorithm is used. When "0", the data compression is not carried out.

[0033]

The directory data is read out in advance from the directory 32 of a tape medium and saved in the buffer memory 14 for reading. When a new file data is recorded or overwritten, the directory data in the buffer memory 14 is duly updated. After the new file data is stored, the updated directory data is saved in the buffer memory 14 and simultaneously recorded on the tape medium to update the directory 32. Assuming that a new file 3 is added after the existing file 2, the direction data is updated as show in Fig. 7(a). As the file attribute data is "1", the file 3 has been compressed by the first compression algorithm. If the data compression is conducted using the second compression algorithm, the file attribute data indicates "2". If no data

compression is made, the file attribute data is "0". Returning back to Fig. 5, the directory data in the buffer memory 14 is updated at Step 204 indicating a new allocation of the file data recorded on the tape medium. The updated directory data in the buffer memory 14 is then transferred by the action of the buffer memory selector 15 to the record/playback module 16. Upon receiving a command from the controller 17, the record/playback module 16 records the updated directory data on the directory 32 of the tape medium (Step 205). Accordingly, the procedure of recording the file data is completed.

[0034]

A procedure of playing back the file data will be described referring to a flowchart shown in Fig. 8. When receiving a file data readout command from the host computer, the controller 17 directs the record/playback module 16 to read the directory data of a desired file data from the directory 32. On the contrary, the record/playback module 16 actuates the drive mechanism, retrieves the directory data from the directory 32 of the tape medium, and transfers it to the buffer memory 18. The buffer memory 18 thus receives the data shown in Fig. 7(a) (Step 301). It is now assumed that the command for reading out the third file data is given by the host computer. The controller 17 advances the tape medium to the file mark located just after the preceding (that is the second one) file data and provides the record/playback module 16 with a command for reading the file data before the mark of the succeeding file data. In response, the record/playback module 16 reads out and transfers

the data to the buffer memory 18 (Step 302).

[0035]

Upon acknowledging from "1" of the file attribute data 43 that the third file data received has been compressed by the first compression algorithm, the file attribute data detector 22 releases a command for directing the data restoration method switch 23 to turn the data restorer 19 on and the data restorer 20 off and simultaneously directs the buffer memory selector 24 to select the buffer memory 21 (Step 304).

[0036]

In response to the command from the file attribute data detector 22, the data restoration method switch 23 controls the turning on and off of the data restoring action of the two data restorers 19 and 20. As described above, the data restorer 19 is by now turned on while the data restorer 20 remains turned off. This allows the compressed data saved in the buffer memory 18 to be dispatched to the data restorer 19 where it is restored to its original non-compression form before transferred to the buffer memory 21. As the action of the two data restorers 19 and 20 is an inverse of the data compressing action of the two data compressors 4 and 6, its related data restoration algorithm is explained in no more detail. The restored data received by the buffer memory 21 is then delivered by the action of the buffer memory selector 24 via the input/output module 1 to the host computer (Step 305). The remaining of the file data is equally subjected to the data restoration of the data restorer 19 before delivered to the host computer (Step 306).

[0037]

When the file attribute data 403 is "2" indicating that the file data has been compressed by the second compression algorithm, the procedure goes to Step 307 shown in Fig. 9 where the file attribute data detector 22 releases a command for directing the data restoration method switch 23 to turn the data restorer 19 off and the data restorer 20 on and simultaneously, directs the buffer memory selector 24 to select the buffer memory 21. The compressed data saved in the buffer memory 18 is dispatched to the data restorer 20 where it is restored before transferred to the buffer memory 21. The restored data received by the buffer memory 21 is then transferred by the action of the buffer memory selector 24 via the input/output module 1 to the host computer (Step 308). The remaining of the file data is equally subjected to the data restoration of the data restorer 20 before delivered to the host computer (Step 309).

[0038]

In case that the file attribute data 403 is "0" indicating that the file data has not been compressed, the procedure goes to Step 310 shown in Fig. 9 where the file attribute data detector 22 releases a command for directing the data restoration method switch 23 to cancel the data restoring action of both the data restorer 19 and the data restorer 20 and simultaneously, directs the buffer memory selector 24 to select the buffer memory 18. The compressed data saved in the buffer memory 18 is directly transferred by the action of the buffer memory selector 24 via the input/output module 1 to the host computer (Step 311). The

remaining of the file data is equally delivered to the host computer without subjecting to the data compression (Step 312).

[0039]

As described above, the data compression can automatically be carried out with the use of the most effective compression method with no need of the user examining which of the data compression methods is used for implementing the best effect of the data compression, hence allowing the data to be transmitted in or out and recorded at higher efficiency. Also, in the playback mode, the file data can automatically be restored by the most effective method with no need of the user examining which of the data compression methods has been used.

[0040]

A procedure of separating the file data into blocks will be described referring to Figs. 1 and 2. The description starts with recording of a file data in conjunction with a flowchart shown in Fig. 10. The file data from a host computer is received by the input/output module 1 and transferred via the file data separator 2 to the buffer memory 3 (Step 401). The file data separator 2 including a counting means for measuring the data length measure the length of the file data while passing the file data to the buffer memory 3. Simultaneously, upon a predetermined length of the data having been passed, the data separator 2 generates and releases a end-of-block signal to the compression rate comparator 11 (Steps 402 and 403).

[0041]

As previously described, the counter 8 measures the number

of bytes in the data length of the file data upon being saved in the buffer memory 3. Simultaneously, the file data saved in the buffer memory 3 is dispatched to the data compressor 4 and the data compressor 6 for the data compression. The data compressed by the data compressor 4 is saved in the buffer memory 5 and its length is measured by the counter 9. The length is measured by the counter 10 (Step 404) similarly as the data compressed by the data compressor 6 is saved in the buffer memory 7.

[0042]

Assuming that the length of each block in this example is 16 kilobytes, the file data separator 2 releases the end-of-block signal upon receiving 16 kilobytes of the data from the host computer. When receiving the end-of-block signal, the compression rate comparator 11 compares measurements of the length of the non-compressed and compressed data determined by the counters 8, 9, and 10 (Step 405). When the length of the compressed data measured by the counter 9 is the smallest, it is determined that the first compression algorithm is the most effective for the data compression. Therefore, the procedure goes to Step 406 shown in Fig. 11 where the compression rate comparator 11 directs the buffer memory selector 15 to select the buffer memory 5. When the length of the compressed data measured by the counter 10 is the smallest, it means that the second compression algorithm is the most effective for the data compression and then, the compression rate comparator 11 directs the buffer memory selector 15 to select the buffer

memory 7 (Step 409). When the length of the non-compression data measured by the counter 8 is the smallest, it means that both the first and second compression algorithms are not effective for the data compression and then, the compression rate comparator 11 directs the buffer memory selector 15 to select the buffer memory 3 (Step 412).

[0043]

Simultaneously, the compression rate comparator 11 releases a signal for directing the file attribute data generator 13 to determine whether the data of each block is compressed using the predetermined data compression algorithm. The file attribute data generator 13 assigns a code data of the compression algorithm employed for the data compression to the compression attribute data of the block in the directory data of the target file (Steps 407, 410, and 413).

[0044]

An example of the directory data for separation of the file data into blocks is shown in Fig. 7(b). The directory data comprises a number of tables, each table composed of 12 bytes and provided for each block. The table for each block includes a file number data 44 (3 bytes), a block number data 45 (4 bytes), a block size data 46 (4 bytes), and a block attribute data 47 (1 bytes). The block attribute data 407 is a code data of the data compression algorithm employed for compressing the file data. When the block attribute data 47 is "1", the first compression algorithm is used and when "2", the second compression algorithm is used. When "0", the data compression

is not carried out.

[0045]

When the first data compression algorithm is selected, the data is transferred by the action of the buffer memory selector 15 from the buffer memory 5 to the record/playback module 16 for storage on a tape medium (Step 408). When the second data compression algorithm is selected, the data is transferred by the action of the buffer memory selector 15 from the buffer memory 7 to the record/playback module 16 for storage on a tape medium (Step 411). In case that it is determined that the data compression is not effective, the data saved in the buffer memory 3 is directly transferred by the action of the buffer memory selector 15 to the record/playback module 16 for storage on a tape medium (Step 414).

[0046]

Then, the procedure goes to Step 415 for checking whether the file data is exhausted. If the file data is not exhausted and its remaining is received from the host computer, the procedure returns back to Step 401 for repeating the same steps (Step 415).

[0047]

The directory 32 on the tape medium is updated by the same procedure of the flowchart shown in Fig. 5 as for the previous case of no separation of data into blocks. A procedure of data restoration is shown in a flowchart of Figs. 12 and 13. In this case, the procedure is identical to that of the non data separation case shown in Figs. 8 and 9 and detailed explanation

will be omitted. In addition, this procedure is differentiated from that shown in Figs. 8 and 9 by the fact that the retrieval of data, the examination whether the data is compressed , and the judgment of the type of the compression algorithm employed are executed on a block-by-block basis (Steps 502 and 503) and repeated until the processing of all the file data is completed (Step 510). As described above, the data compression of each block separated from the file data allows its effectiveness to be determined more precisely than the front end of the file data to be examined, hence improving the efficiency of the data transmission or the data recording.

[0048]

A second example of the present invention will now be described. As its object is identical to that of the first example, the second example will be explained for differences from the first example. Fig. 14 is a block diagram of a data record/playback apparatus including a data compression controlling apparatus of the second example. Fig. 15 is a block diagram of the data record/playback apparatus showing a data restoration controlling apparatus of the second example. Like block components in the data record/playback apparatus with the data compression controlling apparatus are denoted by like numerals as those of the first example and detailed explanation will be omitted. This example is differentiated from the data record/record apparatus shown in Fig. 1 by the fact that the file attribute data generator 13 is replaced with a block attributed data generator 25. The

block attributed data generator 25 generates the management data of each block and saves it as a block header in the buffer memory 14. The block header is also recorded together with its block data on a tape medium.

[0049]

Like components shown in the data record/playback apparatus with the data restoration controlling apparatus of Fig. 15 are denoted by like numerals as those of the first example shown in Fig. 2 and will be explained in no more detail. This example is differentiated by the fact that the file attribute data detector 22 is replaced with a block attributed data detector 26. The block attributed data detector 26 detects the attribute of each block from its block management data read out from the tape medium and supplied by the buffer memory 18. The second example is particularly designed for providing the same advantages as of the first example without assignment of the directory 32 to the tape medium for carrying the directory data.

[0050]

A procedure of examining the compression of the data of each block separated from a file data in the data compression controlling apparatus and the data restoration controlling apparatus arranged as above described will now be explained. While a procedure of examining the compression of the data of each file is substantially identical to the above procedure with the block header replaced by a filer header, the description involves only the procedure for the data of each block. A procedure of recording a block data is first explained referring

to a flowchart shown in Fig. 16.

[0051]

Primary steps are substantially identical to that of the first example shown in Figs. 10 and 11 for separating a file data into blocks. Steps 601 to 604 are identical to Steps 401 to 404 where the file data received from a host computer is separated into blocks which are then subjected to the data compression and measured of their data length. Also similarly, the compression rate comparator 11 compares measurements of the length of the non-compression data and the compressed data determined by the counters 8, 9, and 10 to examine the effectiveness of the data compression (Step 605). When the length of the compressed data measured by the counter 9 is the smallest, the first compression algorithm is selected for use. When the length of the compressed data measured by the counter 10 is the smallest, the second compression algorithm is selected. In case that the length of the non-compression data measured by the counter 8 is the smallest, it is determined that both the first and second compression algorithms are not effective for the data compression.

[0052]

Also simultaneously, the compression rate comparator 11 provides the block attribute data generator 25 with a signal indicating which of the compression algorithms is used and whether the compression of the block data is executed. The block attribute data generator 25 assigns the block header including a code data of the compression algorithm employed as

the block compression attribute data into the buffer memory 14 (Steps 606, 609, and 612) and directs the buffer memory selector 15 to transfer the block header data to the record/playback module 16 (Steps 607, 610, and 613).

[0053]

Now a structure of the block header in this example is shown in Fig. 18. The block header consists of 12 bytes including a block header identification data 51 (2 bytes), a block number data 52 (4 bytes), a block size data 53 (4 bytes), a block attribute data 54 (1 byte), and an auxiliary data 55 (1 byte). The block attribute data 54 is a code data of the compression algorithm used for the compression of the block data. When the code data is "1", it denotes that the data compression is carried using the first compression algorithm. When "2", the second compression algorithm is used. When "0", the data compression is not carried out.

[0054]

When the first compression algorithm is selected, the compression rate comparator 11 directs the buffer memory selector 15 to select the buffer memory 5. Accordingly, the compressed block data is transferred by the action of the buffer memory selector 15 from the buffer memory 5 to the record/playback module 16 for storage with its block header on a tape medium (Step 608).

[0055]

When the second compression algorithm is selected, the compression rate comparator 11 directs the buffer memory

selector 15 to select the buffer memory 7. Accordingly, the compressed block data is transferred by the action of the buffer memory selector 15 from the buffer memory 7 to the record/playback module 16 for storage with its block header on a tape medium (Step 611).

[0056]

In case that it is determined that the data compression is not effective, the compression rate comparator 11 directs the buffer memory selector 15 to select the buffer memory 3. By doing that, the non-compression data saved in the buffer memory 3 is directly transferred by the action of the buffer memory selector 15 to the record/playback module 16 for storage with its block header on a tape medium (Step 614). Then, the procedure goes to Step 615 for examining whether all the file data has been processed. If not, the procedure returns back to Step 601 to repeat the same steps.

[0057]

A procedure of playing back a file data will be described referring to a flowchart shown in Figs. 19 and 20. Primary steps are identical to those of the first example for separating the file data into blocks shown in Figs. 12 and 13 and will be explained in no more detail. This example is differentiated from the previous example shown in Fig. 9 by the fact that the step of reading the directory data is eliminated and the block attribute data 54 in the block head read out together with the block data is used for examining which of the compression algorithms is employed for the data compression (Steps 701 and

702).

[0058]

As described above, by providing block header on each block and by holding the data compression data, it is allowed to determine the precise effect of the compression, hence allowing the data to be transmitted or recorded at higher efficiency. Also, as no management over the directory data is needed, this example can be applied to a variety of simple systems and the like.

[0059]

Also, this example is based on, but not limited to, two of the data compression techniques and may implement three or more different compression methods with the use of a more corresponding number of the data compressors, the buffer memories, the counters, and the data restorers. Moreover, while the data compression controlling apparatus and the data restoration controlling apparatus are installed in the data record/playback apparatus throughout the examples, they may be provided in a host computer with equal success. Some of the components including the file data separator 2, the counters 8, 9, and 10, the compression rate comparator 11, the data compression method switch 12, the file attribute data generator 13, the buffer memories 15 and 24, the record/playback module 16, the controller 17, the file data attribute data detector 22, the data restoration method switch 23, the block attribute data generator 25, and the block attribute data detector 26 may be assembled together and implemented as a microcomputer. The

example may be connected to a communication network instead of the data record/playback apparatus, thus contributing to declination in the amount of data to be transmitted.